


## In the Claims

---

1. (Currently amended) A system comprising:

a serial bus to connect devices;

the serial bus is connected to a first device and a second device; and

 a data structure within ~~a~~ the first device, the data structure comprising which has a hierarchy of descriptors, each descriptor identifiable by a descriptor specifier consisting of a list identifier and an object identifier ~~containing at least a first list descriptor and a second list descriptor and at least one entry descriptor, each descriptor containing at least one data structure, and~~ wherein the second device using at least one data structure in accesses a descriptor in the hierarchy using a command containing the descriptor specifier for the entry.

2. (Currently amended) The system of claim 1, wherein one of ~~the~~ a first list descriptor and ~~the~~ a second list descriptor has information about a first list.

3. (Currently amended) The system of claim 1, wherein an unambiguous specification of ~~a data structure~~ descriptor is made through a descriptor specifier.

4. (Original) The system of claim 2, wherein the information about the first list is placed in a second list.

5. (Original) The system of claim 4, wherein the second list has a beginning and an end; and

the information is placed at the end of the second list.

6. (Currently amended) The system of claim 4, wherein the information ~~from~~ for the first list is placed in the second list in an extended information field.

7. (Currently amended) The system of claim 6, wherein the second device accesses the extended\_information field ~~allowing the controller to move backwards in a~~ the hierarchy of descriptors.

BZ  
8. (Currently amended) A method comprising:  
coupling a first device and a second device to a serial bus;  
placing into a data structure a descriptor specifier which specifies an entry, the descriptor specifier consisting of ~~by a list identifier and an object identifier and the data structure comprising a hierarchy of descriptors as entries~~.

9. (Currently amended) The method of claim 8, further comprising:  
opening ~~a the~~ data structure by the first device; and  
reading at least one entry in the data structure by a second device.

10. (Original) The method of claim 8, comprising:  
embedding information about a parent entry within a child list descriptor.

✓  
11. (Cancelled)

12. (Currently amended) ~~The A method of claim 11, further embedding information about a root\_list\_ID within a root list descriptor comprising:~~  
reading an extended information field in the root list descriptor by a controller,  
wherein accessing the extended\_information field allow~~sing the second device~~controller  
to move backwards in a data structure hierarchy.

13. (Currently amended) The method of claim 8, ~~further comprising: wherein placing the descriptor specifier is placed in an extended\_information field.~~

14. (Currently amended) The method of claim 13, further comprising:  
reading the information from the extended\_information field to move backwards in the hierarchy.

B2  
15. (Currently amended) ~~The A~~ method comprising:

using a descriptor specifier that specifies an entry in a descriptor hierarchy by, the descriptor specifier consisting of a list\_ID and object\_ID.

16. (Original) The method of claim 15, further comprising:

opening the descriptor using the descriptor specifier.

17. (Currently amended) The method of claim 15, further comprising:

embedding information about a parent entry within ~~the a list~~ descriptor in the hierarchy-specifier.

18. (Currently amended) The method of claim 15~~7~~, further comprising:

placing the information about the parent entry at an end of a child list.

19. (Original) The method of claim 17, further comprising:

reading the information from an extended\_information field.

20. (Currently amended) The method of claim 19, further comprising:

using a descriptor specifier for opening a corresponding a parent entry.

21. (Currently amended) A method comprising:

using a descriptor specifier consisting of having a field of a list descriptor ID and an object ID to access an entry in a descriptor hierarchy.

22. (Currently amended) The method of claim 21, further comprising:

placing ~~the a~~ parent descriptor info block in an extended\_information field.

23. (Currently amended) The method of claim 21~~2~~, further comprising:

accessing the extended\_information field ~~allowing a controller to move~~ backwards in the descriptor hierarchy.

24. (Original) The method of claim 21, further comprising:  
embedding information about a parent entry within a list descriptor.

B2  
25. (Currently amended) The method of claim 21, further comprising:  
placing ~~the~~ information about ~~the~~ a parent entry in a child list.

26. (Currently amended) A method comprising:  
embedding a parent descriptor info block within a list descriptor, the parent descriptor info block comprising a descriptor specifier consisting of a list identifier and an object identifier.

27. (Currently amended) The method of claim 26, further comprising:  
placing ~~a~~ the descriptor specifier ~~info block~~ for the parent descriptor in one of a root list descriptor and a child list descriptor.

28. (Original) The method of claim 27, wherein the root list has a first position and a second position; and  
the descriptor specifier is placed at the second position.

29. (Original) The method of claim 27, wherein the child list has a third position and a fourth position; and  
the descriptor specifier is placed at the fourth position.

30. (Currently amended) The method of claim 27, further comprising:  
using ~~a~~ the descriptor specifier in a descriptor command for opening ~~a~~ the parent entry.

31. (Currently amended) The method of claim 30, further comprising:  
navigating descriptors in a descriptor hierarchy using descriptor specifiers,  
wherein navigating is in a backward direction.

32. (Currently amended) A method comprising:

using a delete descriptor command to delete a child list descriptor in a descriptor hierarchy, the delete descriptor command ~~is configured to~~ also update data in a corresponding parent descriptor and in a corresponding parent list descriptor and delete one of a root list, a child list, and an entry when the child list descriptor is deleted.

33. (Currently amended) The method of claim 32, further comprising:

deleting a child\_ID in ~~a~~ the descriptor hierarchy.

34. (Original) The method of claim 33, further comprising:

updating has\_child\_ID attributes.

35. (Original) The method of claim 34, further comprising:

updating entry\_descriptor\_length.

36. (Original) The method of claim 35, further comprising:

updating list\_descriptor\_length.

37. (Currently amended) The method of claim 32, further comprising:

deleting a child list descriptor in ~~a~~ the descriptor hierarchy.

38. (Currently amended) The method of claim 37, further comprising:

deleting ~~a~~ the corresponding parent entry descriptor.

39. (Original) The method of claim 38, further comprising:

updating no\_of\_\_entry descriptors.

40. (Original) The method of claim 39, further comprising:

updating a list\_descriptor\_length.

41. (Currently amended) The method of claim 40, further comprising:  
deleting a first child list descriptor in ~~a~~ the descriptor hierarchy.

B2  
42. (Original) The method of claim 41, further comprising:  
deleting a second child list descriptor.

43. (Original) The method of claim 42, further comprising:  
deleting a list descriptor.

44. (Currently amended) A system comprising:  
a serial bus to connect devices;  
the serial bus is connected to a first device and a second device; and  
a data structure within ~~a~~ the first device ~~which has~~, the data structure comprising  
a hierarchy of descriptors containing a child list descriptor, wherein ~~which is deleted by~~  
~~the second device~~, the second device uses a delete descriptor command to delete the child  
list descriptor, the delete descriptor command ~~is configured to also delete~~ the child ID  
and one of a root list, a child list, and an entry, and update one of a has\_child ID  
attribute, an entry\_descriptor\_length, and the list\_descriptor\_length when the child list  
descriptor is deleted;  
~~the second device deletes a child\_ID; and~~  
~~one of a has\_child\_ID attribute, an entry\_descriptor\_length, and the~~  
~~list\_descriptor\_length is updated.~~

---